HackNY Application 2015

Sakib Jalal, Rutgers University, Computer Science + Algorithms Concentration, Twitter @sakib_jalal, Github https://github.com/sakib, LinkedIn http://linkedin.com/in/sakibj, Personal Site https://sakib.github.io

----

Technologies proficient in and enjoy working with:
-> Data science, Rails, Python, Java, SQL, MongoDB

Technologies familiar with and would be open to working with:
-> JavaScript, front end development, Ruby, Node.js, C++

Tell us about a time you built something awesome in code. How did you choose it? Why did you enjoy it?

-> https://github.com/sakibj/IronMyo - Iron Myo, PennApps Winter 2015 Project
My two friends and I decided to tackle a virtual reality project, so we built an Oculus Rift IronMan simulator at PennApps last winter. We integrated three different technologies that we had never touched before: Myo armbands, the Oculus Rift development kit and the Unity Game Engine. We learned to use C# to program custom events and design a 3D environment in Unity. After some configuration, we displayed the Unity environment in the Oculus Rift. We then sent arm motion and hand gesture data from two Myo armbands to our Unity game's controller functions, where two conjured arms mirrored the movement of the wearer. You would aim your arms at floating targets within vision and then open your palm to fire repulsor blasts, which caused targets to explode upon collision.

-> We also used the Bloomberg API to display the stock price of Stark Technologies inside the environment, which was difficult to integrate because the Unity engine blocks external HTTP requests without a paid plan. We hacked around this by using C# to call an external Python script, which handled our API requests and returned relevant data to the game controllers.

-> Pictures of IronMyo can be found at its Github repository linked above. My team members and I had never done anything with virtual reality before, so we spent hours debugging what seemed like simple tasks: orienting the virtual arms correctly, calibrating the Myo sensors, detecting game object collisions, parsing external API requests, and many more. But it was so awesome to combine these distinct technologies into one immersive experience, and to see a virtual world actually reflect data from the real world. I had so much fun working with my team to piece together the puzzle that was our PennApps project.

Why is hackNY right for you?

-> Many of my friends and mentors are Rutgers HackNY alumni, and they told me about their transformative experience at the Fellows program where they had the opportunity to connect with brilliant minds and contribute to real projects at startups.

-> This past summer I worked at Icepiration, a Rutgers new media startup, where I helped lead a team of 6 developers, designed and implemented databases with senior software engineers and created a JSON

REST API for a new media platform. I also created a 200-slide deck that was presented over 2 days at our tech/entrepreneurship bootcamp in order to introduce youth to business, computer science and app development. It included presentations ranging from fundamental concepts like loops and objects to the lifecycle of data in distributed applications. I was already interested in tech startups, but the awesome experience I had at Icepiration really inspired me to pursue startup development further. I even used the knowledge I gained to start work with some fellow students on a business venture to develop an iOS application that digitizes business cards. I believe the strong mentoring environment at the Fellows program can prepare me to better play an integral role in the buildup of a fast-growing business that can help improve the lives of countless people.

Tell us about what you hope to learn this summer.

-> I will sharpen my skills while exploring new languages and paradigms of development, especially those relevant to efficiently building up the software infrastructure of a startup. I hope to work on interesting projects and learn about systems and networking at the lowest level. I also want to learn about and implement more sophisticated data structures and algorithms as they apply to real world problems that improve service for a wide audience.

Is there a particular technology or industry you're currently interested in? How come? Where do you see it heading in the future?

-> My areas of interest vary, but I’m particularly interested in data science and the Internet of Things as they apply to the biotech and education sectors. I find database design, analysis and management almost addicting, especially with the sheer amount of information available for extraction today. I love working with people and improving their qualities of health and education, so I really care about the efficiency of our medical and education systems and I believe that they will improve with the advancement of biotech and edtech.

-> I know from experience that modern technology can greatly improve the American education system. For example, I teach a CS111 (Intro to CS) recitation at Rutgers and I communicate with all my students 24/7 on a Slack channel, which is already instilling a strong sense of community and mutual support in my section. I also teach some weekly “Hacker Hour” talks at Rutgers for the Undergraduate Student Alliance of Computer Scientists, where a planned series of online resources can go a long way. But converting this into a distributed system is definitely a more complex issue that I hope to work on in the future.

-> I briefly worked with a surgeon and medical students at the Robert Wood Johnson Medical Center, who showed me how inefficient some hospital systems were: they told me they sometimes didn’t know where their patients were, and asked me whether GPS tracking was a good solution to this, and they navigated a series of dated connection pipelines to access basic patient information. Back when I was in high school, I attended EurekaFest, a week-long innovation conference at MIT, where I watched a couple of professors present their new invention. It was a $30 device that could handle infant respiratory problems just as effectively as its $3000 counterpart in hospitals in the developed world, and they partnered up with hospitals in Africa to implement this breakthrough. I just think this is one of the many simple ways that biotech can revolutionize the medical industry once more people can work on it. Though I do understand that there are heavy regulations on patient privacy which bar the hacker community from implementing technologies in the medical space.

Discuss your technical skills/proficiencies/languages and experience

-> I have written university-scale web apps and JSON APIs using Python, Flask, MySQL and MongoDB. In my CS courses, I have used Java to implement data structures and algorithms to deal with problems like recursively evaluating mathematical expressions with stacks or modeling Facebook friendship graph algorithms to identify connected components and connecting vertices. I have also developed secure multi-threaded banking systems in C (https://github.com/sakibj/CS214/tree/master/bank_server). I have taught basic C++ to some of my students through tutoring websites like InstaEDU. I've used HTML/CSS/JS to code websites, and I used Ruby on Rails to build the HackRU Fall 2015 website. I have designed and implemented databases in MySQL, PostgreSQL and MongoDB, transforming ER Diagrams into reliable data storage. I have configured Apache to work with static websites, Flask apps and Rails apps on Linux servers. I'm familiar with using Github and Phabricator as Version Control Systems, and have some hands-on experience with Bootstrap CSS, MVC Patterns and networking (managing Internet at HackRU). As a Student Systems Programmer, I have experience in Linux Systems Administration (Red Hat, Debian) and bash scripting, as well as with open source package management, particularly using the Rutgers implementation of Red Hat's Koji.

When you're not coding, what do you like to do?

-> When I'm not coding, I like to attend interesting events on or off campus, read about current events in the news or through my newsletter subscriptions (StartUp Digest, TechCrunch) and spend time talking with my friends, students and mentors. I sometimes give talks or help underclassmen with their projects at the CAVE, the hub of the Rutgers CS community. I'm also a member of Chi Psi Fraternity, a social Greek organization on College Avenue Campus, where I like to spend time with the brothers and cook meals with our chef.

I'm the Treasurer of USACS, the Undergraduate Student Alliance of Computer Scientists, where I manage our organization's $70,000 in funds and allocate resources to our myriad of coding and educational events, which I help teach. I helped organize HackRU Fall 2015, where I managed the website and database of 2000 registrants, and also played a critical role in distributing Internet and Food to 1000 developers for 30 hours.

I also really like going on last-minute adventures with my close friends. Every now and then, my friends and I decide at midnight that we want to go to Philadelphia or New York, and we just drive there and enjoy the nightlife. Or just before the weekend, we say we want to go on a hike, drive up to nearby mountains with camping gear and spend time with each other.

What sectors would you especially like to work in? (e.g., fashion, data science, media, art, education...)

-> Data science, biotech, and education

What sectors would you prefer not to work in?

-> Art, fashion, government

List five NYC startups you'd love to work at. If you're accepted, we'll do our best to match you with one of your favorites. If they're unavailable, we'll use this list to help make a great startup match for you.

->
Oscar
Schoology
Stack Overflow
Artemis-health
Days-by-wander

Any free text you'd like us to add when describing you to proposed host startups?

-> Over the past summer, I closely followed Y-Combinator President Sam Altman's Stanford lecture series on "How to Start a Startup" and applied the knowledge I gained to a few startups at Rutgers that I mentioned above. I found the distinct stories of founders and CEOs of successful startups to be both interesting and inspirational, and I look forward to having the opportunity to learn and share stories of my own after HackNY.

What size startup would you like to work with (total number of employees)?

1-5
6-15
16-25 <- this one
26-50
51+

Paste in a ~100 line sample of code you're proud of.*

-> Shrunk appserver and database implementations. 3 Code snippets
http://pastebin.com/cYYkkWxY

Tell us why you wrote this code and what it does.*
-> The above code is from the back-end of Go, the Rutgers Official URL Shortener. The project name is "Shrunk." I will explain what each of the 3 snippets above do and why.

I. Links are stored as Python dictionaries in the URL Shortener. Users may request to sort them by alphabetical or reverse alphabetical order. If so, we need to run specific MongoDB cursor operations. In order to interact with a set of results returned from a query in MongoDB we need a cursor, and the unfortunate thing is that if you move forward with the cursor, you cannot backtrack on the results. So we clone the cursor into page_cursor to retain the original cursor's position before paginating page_cursor (splitting the results into distinct pages), which is implemented with the paginate() method defined elsewhere and the MAX_DISPLAY_LINKS application variable defined in config.py. We then call Python's built in sorted method which takes in a set of links returned from cursor.get_results() and the key to sort by, which is specified as the lowercase string-casted field "title" of the link dict. This line does the brunt of the work, and if the user requested reverse alphabetical order, we reverse this sorted list. We then return the set of links that corresponds to the current "page" that the user is browsing by choosing the correct offset of the DB query results, based on MAX_DISPLAY_LINKS.

II. This is essentially a one-liner that takes advantage of Python's built-in functions to build a factory of methods passing results to other methods. The goal is to append a field "ru_visits" to each link dictionary that represents the number of IPs from Rutgers that have visited the given link. It makes sense to read this from the inside to the outside. We use the custom-built database client method "get_visits" to get a list of all the visit objects that have visited a single link. We extract the source_ip from each of these visit objects with map(), then convert it back to a list for the next method. Now that we have all the visitor IPs for a given link, we filter() out the IPs that are not from rutgers using a lambda that keeps only source_ip's that begin with any of the valid Rutgers IP addresses, defined in config.py as RUTGERS_IP_LIST. This filter is turned into a list of Rutgers IPs that have visited a given link, which is then passed to len() to get the number of unique Rutgers visitors and stuck into the "ru_visits" field of said link dictionary. This is done for all link in links. What I find elegant in this is that it's essentially one line of code, and though I tried I couldn't find a way to get rid of the loop iterating over all links. I made sure to organize it by tabs and new lines to keep it readable.

III. This snippet is from the client.py file which abstracts all database client interactions. Essentially, when we want to run a database query, in the case that the query fails, the database goes down or something else goes wrong, the application gracefully redirects to an error page and contacts the admins. The ShrunkClient object is the client that the application uses, and in most cases, its __init__() returns in the try statement. If the DB is down, it sleeps and tries again a few more times before returning with a null connection, which the app interprets as a database error. The client itself passes a MongoDB connection object to MongoProxy, a class that wraps the client in a shell in which all valid EXECUTABLE_MONGO_METHODS (defined at the top) are executed through another wrapper called ShrunkExecutable, which takes a given method and executes it with the safe_mongocall decorator. This decorator tries to execute a method 5 times before returning nothing after consistent connection errors. This elaborate setup ensures a level of abstraction that is flexible insofar that any method can be added to EXECUTABLE_MONGO_METHODS, yet stable enough to gracefully handle actual connection errors. This has been tested in edge cases where the database goes down or starts up while the query is being executed and works as expected.

Please provide links and descriptions of two of your favorite coding projects in any language.*

1. https://github.com/oss/shrunk Go: Rutgers Official URL Shortener
My coworkers and I maintain this Git Repository for the Rutgers URL Shortener, currently in production on http://go.rutgers.edu. The project name is "Shrunk" but its deployment is called "Go." My coworkers and I built this with Python, Flask and MongoDB for the database. So far, I've implemented database stability, detailed logging for all routes and actions, source IP tracking per URL and alphabetical URL sort (check my commits!). The templates are written with Jinja2, a templating language for Flask applications.

Generally, the application is modular insofar that different files handle different aspects of the app. For example, client.py serves to abstract all of the database interactions (inserts, updates, deletes) from the application itself, so that those developing the appserver can call a client object and not worry about its internals. My commits are at https://github.com/oss/shrunk/commits?author=sakibj

2. https://github.com/RutgersRoboticsResearch/HackRU-Fall-2015 - HackRU Fall 2015 Web Application
As an organizer of HackRU Fall 2015, I collaborated with my friend Jon to build, host and maintain the Rails application for hackru.org. We chose to use Rails because of its well-documented capabilities and scalability as a web framework. I also really wanted to get experience with Ruby, Rails, hosting web servers and managing databases. It was like a hackathon project - long hours of digging through documentation, day after day of changing things based on user feedback, and yet despite the toil I was still excited to learn more about Rails custom controllers, before/after filters and the like. When the server went down or the application had issues, I got angry emails and dropped what I was doing to fix it (is this what Software Reliability Engineers feel like?). There was a lot of front-end and MVC pattern development involved, but my favorite parts during development included the Ubuntu server setup of the Digital Ocean development and production droplets, Apache configuration with Passenger and management of the PostgreSQL database. I especially enjoyed writing database backup scripts, spinning up SQL queries and just monitoring the growing database of users to pick out interesting trends in the data.

3. https://github.com/sakibj/leapwheels - Leap Wheels, Dragon Hacks Winter 2015 Project
I went to Dragon Hacks at Drexel University last winter and met some cool Drexel juniors who worked with me on our project: a car that moves according to your hand gestures. We built the car from scratch using cardboard, toy wheels, tape and 3D printed axles. We added motors and an Intel Edison board that served to wirelessly read in serial input generated by a nearby computer on the same network, which was configured to send incoming data from our Leap Motion through a socket to the Intel Edison. I used the python wrapper for the leap motion API to interpret the data, and the Drexel engineering students knew how to configure the board to send commands to an Arduino to turn the motors based on the input that I sent. Here’s a video of it working: https://vimeo.com/117357460. Overall I thought it was a super cool project and I’m lucky to have met those awesome teammates!

4. https://github.com/sakibj/FBeMail/ - FBeMail, PennApps Winter 2013 Project
When I was a senior in highschool, I dragged my friend to PennApps, where we realized that we always keep open one tab of Gmail and one tab of Facebook. What if we just had one tab that showed both? We decided to integrate your Gmail inbox into Facebook, so we wrote a chrome extension in JavaScript to inject code into the body of your Facebook page. The code pulls data from a static gmail page that serves XML, runs it through an XML parser and feeds it into the page under a mail icon that we created. The icon displays next to the standard three at the navbar and when clicked, opens a dropdown list of your email summaries. A screenshot can be seen under Projects at my website http://sakibj.github.io.

Please share anything else you'd like to share about yourself.

-> More of my projects can be found on my Linkedin page - I felt that I should mention the major ones.